

A hand holding puzzle pieces with binary code on a blue background with binary digits.

Порты ввода/вывода

Языки программирования (часть 1)

> Порты ввода/вывода

```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

Прикладная программа



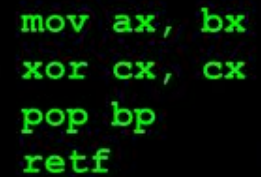
Операционная система



BIOS / UEFI



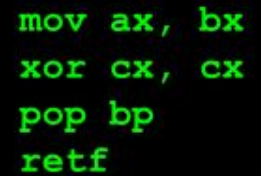
> Порты ввода/вывода



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Используются для взаимодействия с внешними устройствами.
- 2^{16} портов по 8 бит.
 - С номерами от 0000 до FFFF.
 - F8–FF — зарезервированы.
 - Соседние 8-битные порты могут группироваться в 16- и 32-битные.

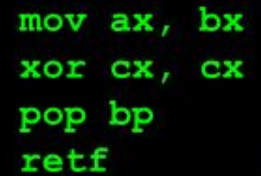
> Инструкция IN



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Операнды:
 - первый (приёмник) — AL/AX/EAX;
 - второй (источник) — imm8 или DX.
- Читает значение из порта.
 - Второй операнд задаёт номер порта.


> Инструкция OUT



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Операнды:
 - первый (приёмник) — imm8 или DX;
 - второй (источник) — AL/AX/EAX.
- Записывает значение в порт.
 - Первый операнд задаёт номер порта.

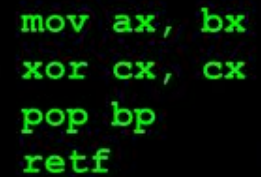
> Инструкции INS и OUTS



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Среди строковых команд есть две для работы с портами:
 - INSB/INSW/INSD
 - OUTSB/OUTSW/OUTSD
- Принцип работы — аналогично другим строковым командам.

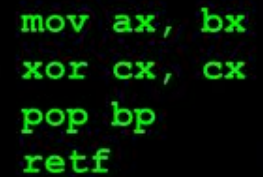
> Порты ввода/вывода



```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

- Внешним устройствам ставятся в соответствие те или иные порты.
 - Для некоторых устройств номера портов фиксированные.
 - Для некоторых — могут изменяться.
- Запись в порт — отправка данных устройству.
- Чтение из порта — приём данных от устройства.

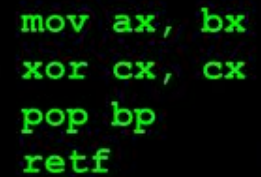
> Порты ввода/вывода



```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

- Пример:
 - Видеоадаптер VGA использует порты с номерами 03B0–03DF.
 - Порты 03C7, 03C8 и 03C9 можно использовать для чтения/изменения используемого набора цветов.

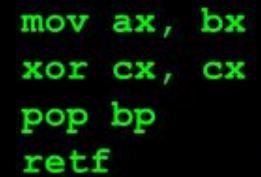
> Порты ввода/вывода



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Изменение цвета номер X (0–255):
 - В порт 3C8 записать номер цвета X.
 - В порт 3C9 последовательно записать
 - красную составляющую (0–63);
 - зелёную составляющую (0–63);
 - синюю составляющую (0–63).

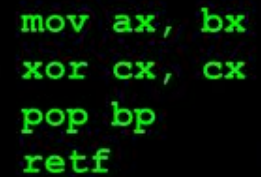
> Порты ввода/вывода



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Получение цвета номер X (0–255):
 - В порт 3C7 записать номер цвета X.
 - Из порта 3C9 последовательно читать
 - красную составляющую;
 - зелёную составляющую;
 - синюю составляющую.

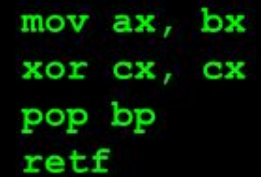
> Порты ввода/вывода



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Порты 0040–0043 используются для управления таймерами.
 - Есть три канала таймеров:
 - Канал 0 отвечает за таймер, срабатывающий 18.2 раза в секунду и генерирующий прерывание 08h.
 - Канал 1 отвечает за memory refresh.
 - Канал 2 позволяет управлять встроенным динамиком (пищалкой).

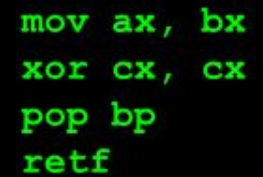
> Порты ввода/вывода



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Порты 0060–0063 используются для управления PPI — Programmable Peripheral Interface.
 - Бит 1 порта 0061 управляет состоянием встроенного динамика:
 - Всего 2 состояния.
 - Изменяя значение этого бита с определённой частотой, можно добиться звука требуемой частоты.

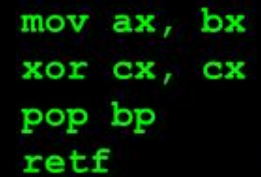
> Порты ввода/вывода



```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

- Для вывода звука можно соединить выход таймера 2 со входом динамика.
 - См. TechHelp -> Speaker control

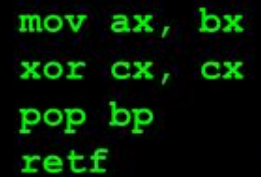
> Порты ввода/вывода



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Порты 0330 и 0331 используются для взаимодействия с MPU-401 (если есть такое устройство).
 - Эмулируется NTVDM и DOSBox.
 - Но в NTVDM есть ошибка эмуляции, из-за которой нота начинает звучать только после записи следующего сообщения.

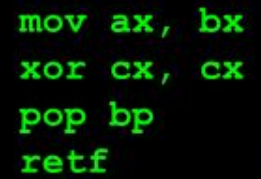
> MPU-401



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- MPU — MIDI Processing Unit.
- MIDI:
 - 16 каналов (канал 10 — перкуссия);
 - один из 128 инструментов в каждом.
- В режиме UART MP-401 просто передаёт поступающие данные на устройство воспроизведения.

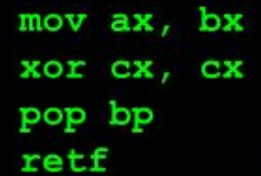
> MPU-401



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Порт 0331 — управляющий:
 - используется для отправки команд и определения готовности MPU-401 к приёму/передаче данных/команд.
- Порт 0330 — порт данных:
 - используется для передачи данных (MIDI-сообщений).

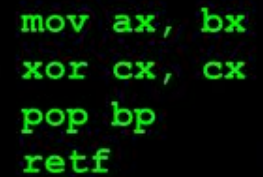
> MPU-401



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Сброс устройства:
 - отправить команду FFh в порт 331.
- Переключение в режим UART:
 - отправить команду 3Fh в порт 331.
- Воспроизведение звуков:
 - посылать MIDI-сообщения в порт 330.

> MIDI-сообщения



```
mov ax, bx
xor cx, cx
pop bp
retf
```

Байт статуса	Байты данных	Описание
1000nnnn	0kkkkkkk 0vvvvvvvv	Note Off
1001nnnn	0kkkkkkk 0vvvvvvvv	Note On
1100nnnn	0pppppppp	Program Change

nnnn — номер канала

kkkkkkkk — номер клавиши (ноты)

vvvvvvvv — сила (скорость) нажатия/отпускания

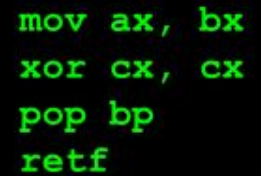
pppppppp — номер программы (инструмента)

> MIDI-сообщения

```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

Нота	Октава										
	-2	-1	0	1	2	3	4	5	6	7	8
C	0	12	24	36	48	60	72	84	96	108	120
C#	1	13	25	37	49	61	73	85	97	109	121
D	2	14	26	38	50	62	74	86	98	110	122
D#	3	15	27	39	51	63	75	87	99	111	123
E	4	16	28	40	52	64	76	88	100	112	124
F	5	17	29	41	53	65	77	89	101	113	125
F#	6	18	30	42	54	66	78	90	102	114	126
G	7	19	31	43	55	67	79	91	103	115	127
G#	8	20	32	44	56	68	80	92	104	116	---
A	9	21	33	45	57	69	81	93	105	117	---
A#	10	22	34	46	58	70	82	94	106	118	---
B	11	23	35	47	59	71	83	95	107	119	---

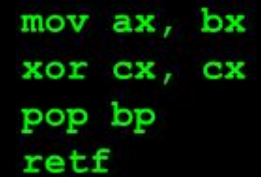
> MPU-401



```
mov ax, bx
xor cx, cx
pop bp
retf
```

- Бит 6 порта 330h равен 0, когда MPU-401 готов принимать данные.
 - Нужно дождаться этого перед отправкой MIDI-сообщения.
- Бит 7 порта 330h равен 0, когда MPU-401 хочет передать данные.
 - Нужно считывать данные перед продолжением работы.

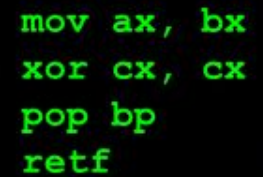
> Порты ввода/вывода



```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

- Для большинства устройств взаимодействие намного сложнее.
 - С ними приходится общаться через контроллер шины, к которой они подключены (ISA, PCI и т.п.).
 - Само устройство может содержать сотни т.н. регистров и поддерживать десятки различных режимов работы.

> Порты ввода/вывода



```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

- Для большинства устройств взаимодействие намного сложнее.
 - Драйвер — набор процедур, реализующих взаимодействие с устройством.
 - ОС определяет, какие процедуры должен содержать драйвер.

> Вопросы?

```
mov ax, bx  
xor cx, cx  
pop bp  
retf
```

